

# Diseño Conceptual

## Objetivos

### Comprender:

- \* **Nociones de diseño de Sistemas de Información**
- \* **Modelo Entidad / Relación**  
**Elementos y reglas**  
**Restricciones**
- \* **Construcción del esquema E / R**  
**Ejemplos**  
**Ejercicios**

Sistemas de Información - Ciclo de vida  
Proyecto clásico



# Sistemas de Información - Ciclo de vida

## Proyecto clásico

### **Estudio de factibilidad:**

- \* Opciones alternativas y costos**
- \* Prioridades**

Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

## **Diseño y planificación:**

- \* Relevamiento de requerimientos y análisis.**
- \* Características y funcionalidades.**
- \* Interacción con usuarios, para definir datos, operaciones, volúmenes y frecuencias.**
- \* Determinación de requerimientos de hardware y software.**

Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

## **Diseño y planificación:**

- \* **Diseño**

  - Datos y aplicaciones.**

  - Descripciones formales, según modelos.**

- \* **Planificación**

  - Fases de trabajo y entregas.**

Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

## **Implementación:**

- \* Construcción y carga de la base de datos.  
Desarrollo de software.**

Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

**Pruebas:**

**De funcionalidades y calidad general.**

**En condiciones operativas, si es posible.**



Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

**Producción y administración:**

- \* El sistema de información está trabajando en condiciones reales.**

**Control y mantenimiento.  
Resguardo.**

Sistemas de Información - Ciclo de vida  
**Proyecto clásico**

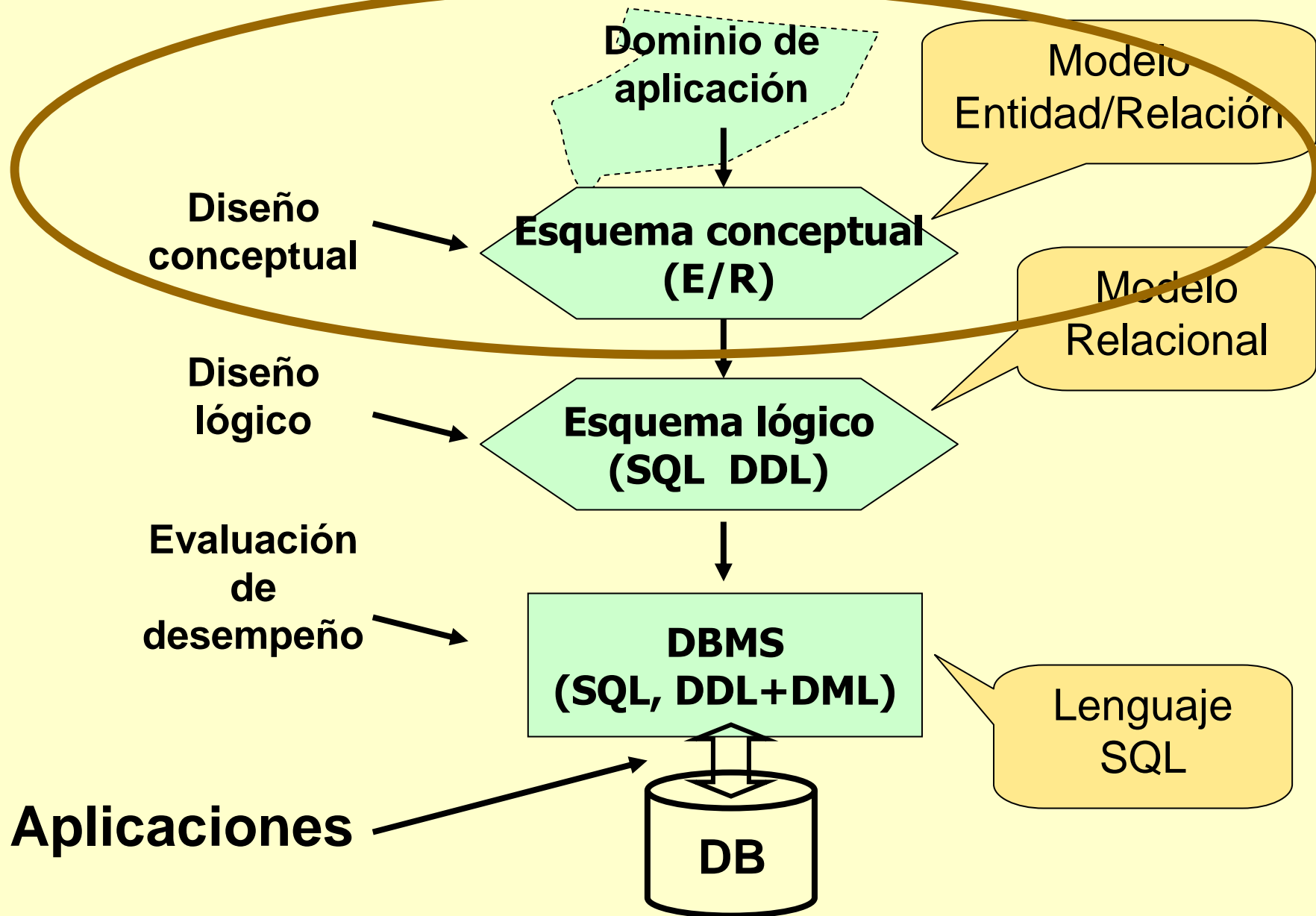
**Evaluación:**

**Problemas de funcionamiento.**

**Nuevos requerimientos.**

**Análisis, diagnóstico, terapéutica.**

# Sistemas de Información - Diseño



## **Relevamiento de requerimientos y análisis :**

**Los usuarios suministran requerimientos heterogéneos y no formales acerca de las operaciones a implementar en el sistema de información.**

- \* Características de los datos.**
- \* Operaciones sobre los datos.**
- \* Restricciones sobre datos y operaciones.**

## **Relevamiento de requerimientos y análisis :**

### **Actividades principales**

- \* Construir el glosario.**
- \* Resolver ambigüedades (sinónimos, homónimos, etc.).**
- \* Agrupar requerimientos homogéneos.**

## **Relevamiento de requerimientos y análisis :**

**No es sencillo seguir una metodología standard en esta fase**

**El objetivo es...**

**entender qué quieren los usuarios.**

## Fuentes de conocimiento

- \* **Usuarios**  
Entrevistas, encuestas, etc.
- \* **Documentación existente**  
Formularios, documentos internos, procedimientos, leyes, normativa, etc.
- \* **Legado**  
Aplicaciones a sustituir.  
Aplicaciones con las que interactuar

## **Ejemplo de requerimiento en lenguaje natural**

**Necesitamos crear una base de datos para cursos. Recopilaremos datos acerca de participantes y profesores. Los participantes (alrededor de 5000) se identificarán por DNI y tendrán Apellido, Nombre, Edad, Sexo, Lugar de nacimiento, Nombre del empleador actual, Lugares de trabajo y Fechas, Dirección, Teléfono, Materias cursadas (hay unas 200) y Título. También representamos las materias que están cursando y, para cada día, las aulas en las que se dan las clases. Los cursos tienen Código, Título, y pueden tener fechas de inicio y fin y cantidad de participantes.**



## **Ejemplo de requerimiento en lenguaje natural**

**Para estudiantes que no trabajan para un empleador, se tiene que conocer el área de interés y el título secundario. Para aquellos alumnos en relación de dependencia se quiere saber el nivel y la posición.**

**Para docentes (alrededor de 300) se recopila Apellido, Nombre, Edad, Lugar de nacimiento, Nombre del curso que dictan, Cursos que dictaron en el pasado y Cursos que podrían dictar.**

**Almacenamos también teléfonos. Quienes enseñan pueden ser empleados o free lance.**

# Reglas generales:

## Evitar ambigüedades

Necesitamos crear una base de datos para cursos. Recopilaremos datos acerca de participantes y profesores. Los participantes (alrededor de 5000) se identificarán por DNI y tendrán Apellido, Nombre, Edad, Sexo, Lugar de nacimiento, Nombre del empleador actual, Lugares de trabajo y Fechas,...

...  
Para docentes (alrededor de 300) se recopila Apellido, Nombre, Edad, Lugar de nacimiento, Nombre del curso que dictan, Cursos que dictaron en el pasado y Cursos que podrían dictar.

# Reglas generales:

## Evitar ambigüedades

Necesitamos crear una base de datos para cursos. Recopilaremos datos acerca de participantes y profesores. Los participantes (alrededor de 5000) se identificarán por DNI y tendrán Apellido, Nombre, Edad, Sexo, Lugar de nacimiento, Nombre del empleador actual, Lugares de trabajo y Fechas, Dirección, Teléfono, Materias cursadas (hay unas 200) y Título. También representamos las materias que están cursando y, para cada día, las aulas en las que se dan las clases. Los cursos tienen Código, Título, y pueden tener fechas de inicio y fin y cantidad de participantes.

# Reglas generales:

## Evitar ambigüedades

Necesitamos crear una base de datos para cursos. Recopilaremos datos acerca de participantes y profesores. Los participantes (alrededor de 5000) se identificarán por DNI y tendrán Apellido, Nombre, Edad, Sexo, Lugar de nacimiento, Nombre del empleador actual, Lugares de trabajo y Fechas, Dirección, Teléfono, Materias cursadas (hay unas 200) y Título...

...

Para estudiantes que no trabajan para un empleador, se tiene que conocer el área de interés y el título secundario....

# Reglas generales:

## Evitar ambigüedades

Necesitamos crear una base de datos para cursos. Recopilaremos datos acerca de participantes y profesores. Los participantes (alrededor de 5000) se identificarán por DNI y tendrán Apellido, Nombre, Edad, Sexo, Lugar de nacimiento, Nombre del empleador actual, Lugares de trabajo y Fechas, Dirección, Teléfono, Materias cursadas (hay unas 200) y Título. También representamos las materias que están cursando y, para cada día, las aulas en las que se dan las clases. Los cursos tienen Código, Título, y pueden tener fechas de inicio y fin y cantidad de participantes.

# Reglas generales:

## **Frases con estructura standard**

... Los participantes (alrededor de 5000)  
se identificarán por DNI y tendrán Apellido, Nombre,  
Edad, Sexo, ....

... Para los participantes (alrededor de 5000)  
se registra DNI (identificador), Apellido, Nombre,  
Edad, Sexo, ....

# Reglas generales:

## Evitar frases involucrantes

Para estudiantes que no trabajan para un empleador, se tiene que conocer el área de interés y el título ...

Para estudiantes no empleados, se tiene que conocer el área de interés y el título secundario.

# Reglas generales:

## Evitar frases involucrentes

Para aquellos alumnos **en relación de dependencia** se quiere saber el nivel y la posición

Para aquellos alumnos **empleados,** se quiere saber el nivel y la posición



## **Reglas generales:**

**Elegir el nivel apropiado de abstracción**

**Separar párrafos referidos a datos de los referidos a operaciones**

**Hacer referencias explícitas a términos.**

**Reorganizar frases que refieren a los mismos conceptos.**

# Interacción con los usuarios

**Es una actividad que requiere un desarrollo cuidadoso.**

- \* Diferentes usuarios pueden suministrar diferente información.
- \* **Los usuarios de alto nivel, tienen generalmente una visión más amplia, pero menos detallada.**

# Interacción con los usuarios

**En general es útil:**

- \* Verificar permanentemente comprensión y coherencia.**
- \* Comprobar por medio de ejemplos:**
  - + Casos generales**
  - + Casos extremos**
- \* Solicitar definiciones y clasificaciones.**
- \* Poner en evidencia los aspectos principales con respecto a casos especiales.**

## Ejemplo de glosario de términos:

<b>Término</b>	<b>Descripción</b>	<b>Sinónimos</b>	<b>Referencia</b>
<b>Alumno</b>	<b>Persona que cursa una materia. Puede ser empleado o autónomo.</b>	<b>Participante Estudiante</b>	<b>Materia Empleador</b>
<b>Docente</b>	<b>Dicta una materia. Empleado o autónomo.</b>	<b>Profesor</b>	<b>Materia</b>
<b>Materia</b>	<b>Cada una de las asignaturas de la carrera.</b>	<b>Curso</b>	<b>Docente Alumno</b>
<b>Empleador</b>	<b>Patrono actual o anterior .</b>	<b>Empresa en la que trabajan</b>	<b>Alumno</b>

# Reestructuración de requerimientos

- \* **Eliminar homónimos**  
**Utilizar un único término para cada concepto.**

# Reestructuración de requerimientos

- **Reorganizar frases agrupando**

*En el ejemplo anterior:*

- \* **Frases generales**
- \* **Frases para alumnos**
- \* **Frases para docentes**
- \* **Frases para materias**
- \* **Frases para empleadores**

# Diseño conceptual

- **A partir de los requerimientos se genera:**
  - \* **Un esquema conceptual**
  - \* **Una descripción formal de los requerimientos.**

# Diseño conceptual

- **A partir de los requerimientos se genera un esquema conceptual:**
  - \* **Una descripción formal de los requerimientos.**
  - \* **Independiente del DBMS.**
- \* **Modelo conceptual:**
  - \* **Permite realizar descripciones de alto nivel, independientes de la implementación.**
  - \* **Algunas posibles elecciones. Standards ampliamente aceptados.**



# Diseño lógico

- Traducir el esquema conceptual en modelo de datos comprensible por el DBMS
- \* El resultado es un esquema lógico, expresado con el DDL.
- \* También se considera:
  - Restricciones.
  - Performance.

# Diseño físico

•Se tiene que tomar en cuenta el DBMS específico:

- Microsoft SQL Server, IBM DB2, Oracle, MySQL, etc.
- El mismo esquema lógico puede tener distintas representaciones físicas, para obtener mejor performance.

# Diseño físico

- **El esquema físico incluye:**
  - **Estructuras de almacenamiento.**
  - **Agrupaciones físicas.**
  - **Métodos de acceso...**

# Modelo de datos:

## Lógico vs. conceptual

- **Un modelo de datos es un conjunto de conceptos utilizados para describir datos, relaciones y restricciones.**

# Modelo de datos:

## Lógico vs. conceptual

### Modelos lógicos

- La estructura de datos entendida por el DBMS
- Utilizados también en programas de aplicación
- Independientes de las estructuras físicas

### Modelos conceptuales

- Representación de datos independiente del software
  - Describe conceptos del mundo real
  - Utilizado en la fase de diseño inicial
- Entidad-Relación
- UML

**Modelo de datos:**

**Lógico vs.  
conceptual**

**Modelos lógicos**

**Modelos conceptuales**

**Es necesario comprender  
las dos fases del diseño:  
conceptual y lógico.**

**Modelo de datos:**

**Lógico vs.  
conceptual**

**Modelos lógicos**

**Modelos conceptuales**

**Se buscará la comprensión desde lo conceptual a lo lógico, en casos simples.**

**Se verán los primeros resultados del diseño y la implementación de estructuras.**

**Se abordarán casos más complejos.**

# **Modelos conceptuales**

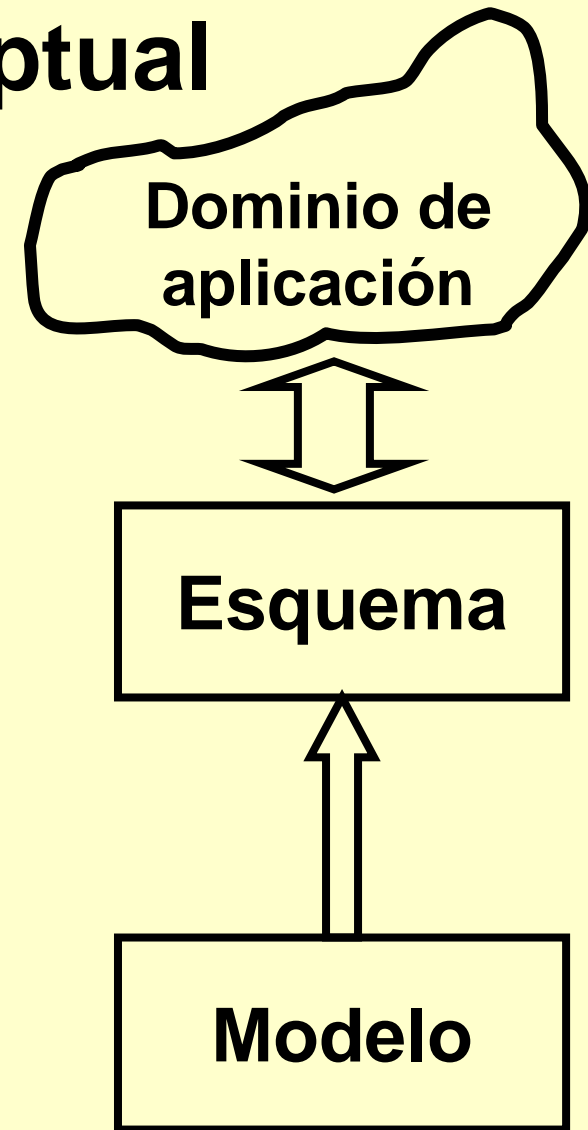
## **Historia**

- \* Modelos semánticos, RM/T (principio de los '70)**
- \* Entidad / Relación (E / R) [P. Chen, 1976 ]**
- \* IDEFIX [ Standard gubernamental, v.g. ERWin]**
- \* UML (Universal Modeling Language) [ 1999 ]**



# El rol del modelo conceptual

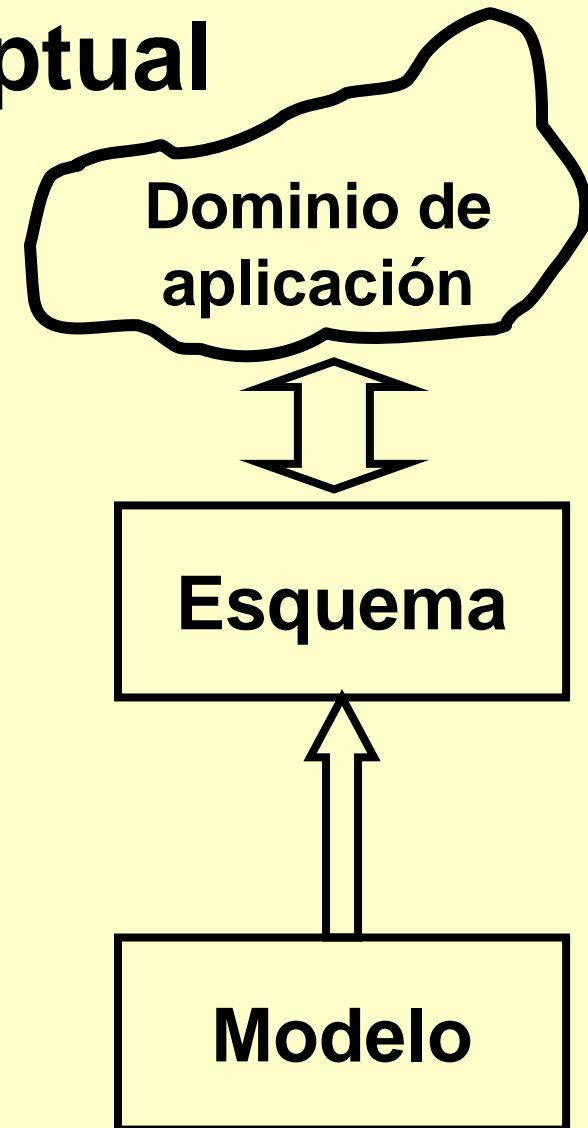
**Se genera un  
esquema que  
represente al  
dominio de  
aplicación,  
independientemente  
del DBMS**



# El rol del modelo conceptual

**El nivel de abstracción es intermedio entre usuarios y software**

- \* Gráfico**
- \* Flexible**
- \* Intuitivo**
- \* Potente**



# Modelo E / R

## Conceptos principales:

- \* Entidad.
- \* Relación.
- \* Atributo.

## y también:

- \* Restricciones de cardinalidad.
- \* Identificadores.
- \* Jerarquía.

## Modelo E / R

### Entidad:

Conjunto de objetos en el dominio de aplicación, con características comunes (por ejemplo: personas, autos, etc.) y con existencia autónoma.

## Modelo E / R

### Entidad:

Una entidad tiene como elementos, objetos específicos (por ejemplo: yo, mi auto, etc.).

# Modelo E / R

## Entidad:

La representación gráfica de una entidad es el rectángulo con un nombre adentro.

**Persona**

**Auto**

**Empleado**

## Modelo E / R

### Relación:

Es el vínculo lógico entre entidades.

## Modelo E / R

### Relación:

Tiene como elementos una agregación de elementos de las entidades.



## Modelo E / R

**Relación:**

Se representa con un rombo.



## Modelo E / R

### Relación:

Si **p** es un elemento de Persona y **c** es un elemento de Ciudad, el par **(p,c)** puede ser un elemento de la relación *Vive en*

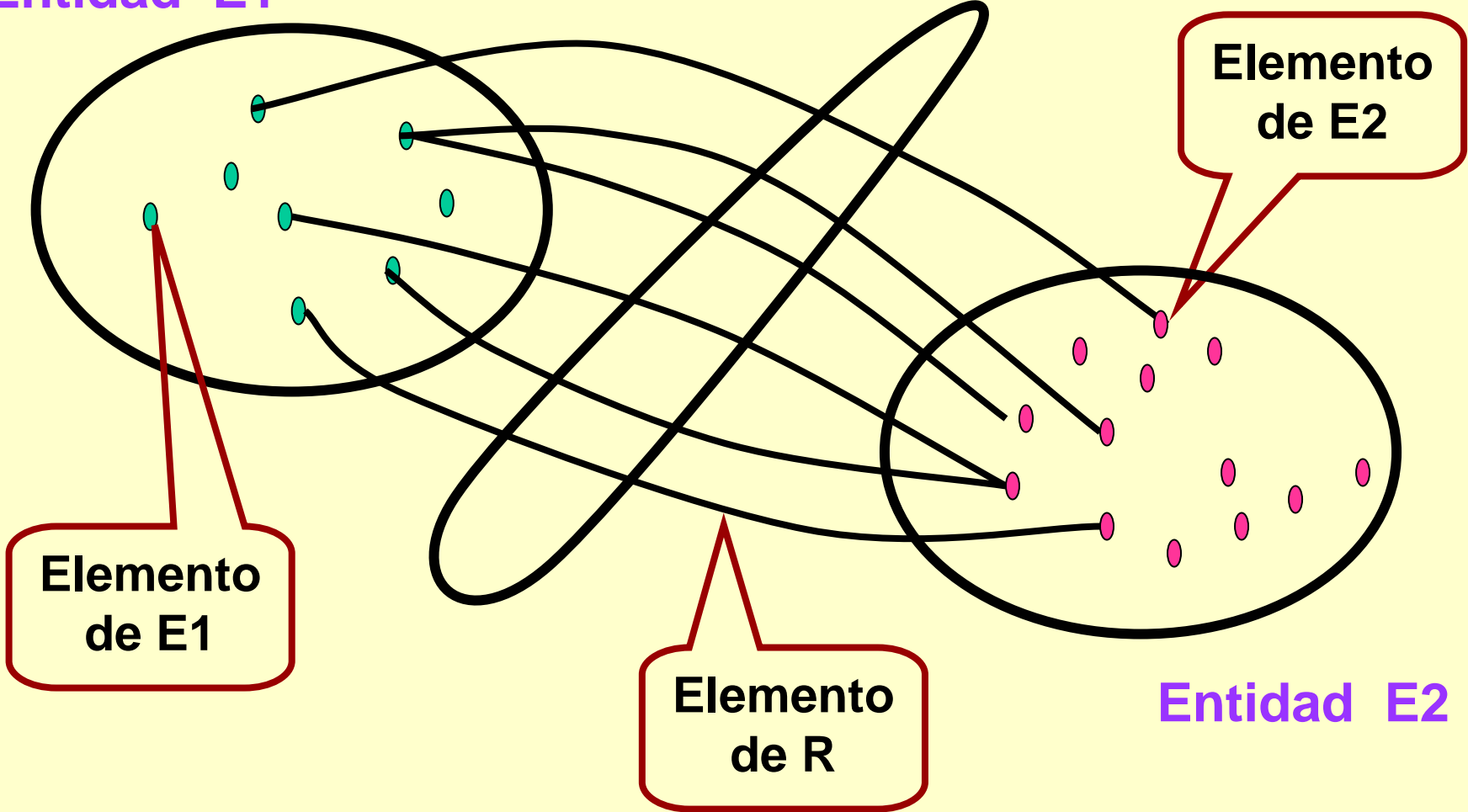


# Modelo E / R

Nivel elemento

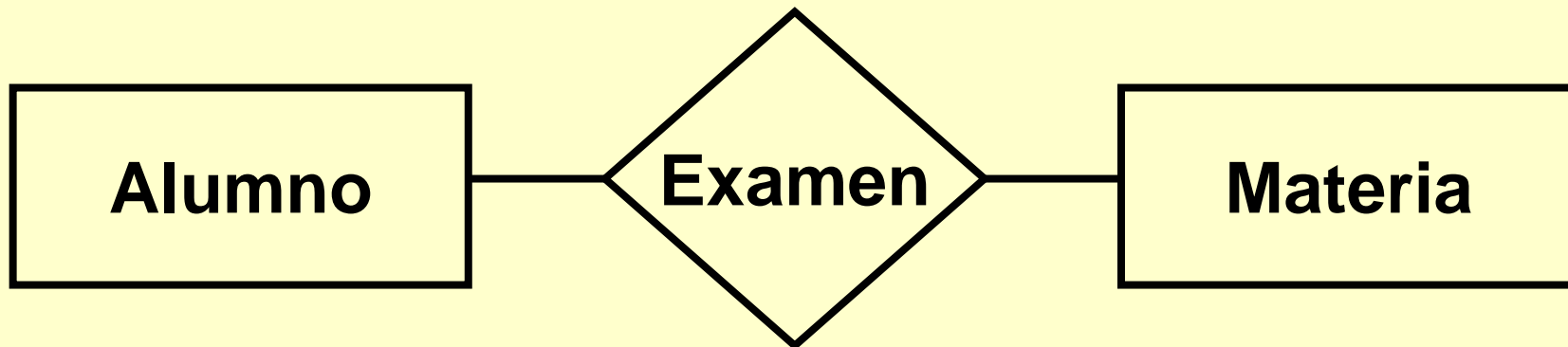
Entidad E1

Relación R

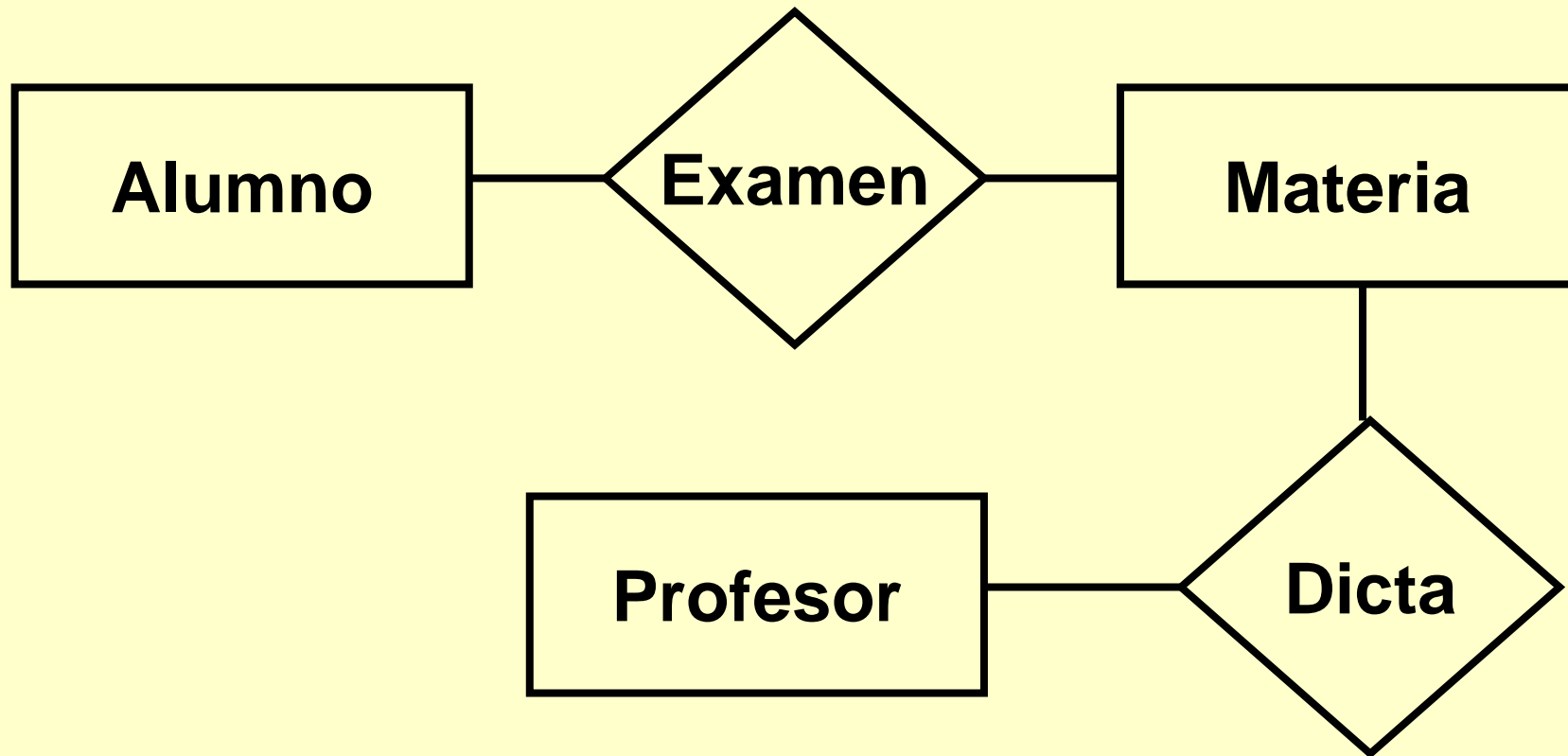


# Elementos de las asociaciones

- \* **Conjunto de posibles elementos:**  
El producto cartesiano del conjunto de elementos de las entidades participantes.
- \* **Sin elementos duplicados.**
- \* **Si  $a$  es un alumno,  $m$  una materia, el par  $(a,m)$  puede aparecer sólo una vez en la relación Examen.**



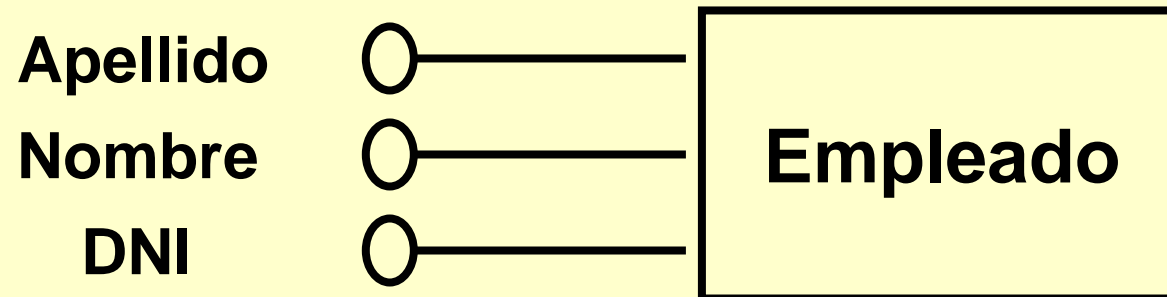
# Un esquema simple:



# Atributo:

Es una propiedad elemental de una entidad o una relación.

Una representación:

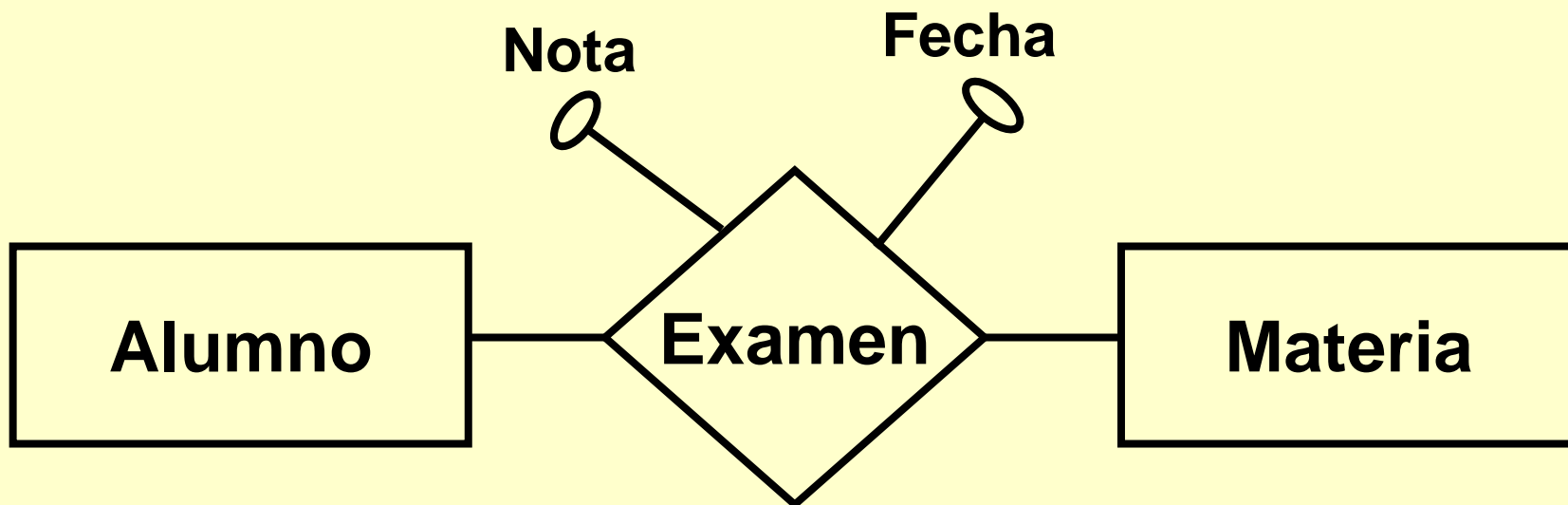


Apellido, Nombre, DNI, son atributos de Empleado.

## Atributo:

¿Se asocia a una entidad o a una relación?

¿Dónde poner Nota y Fecha?

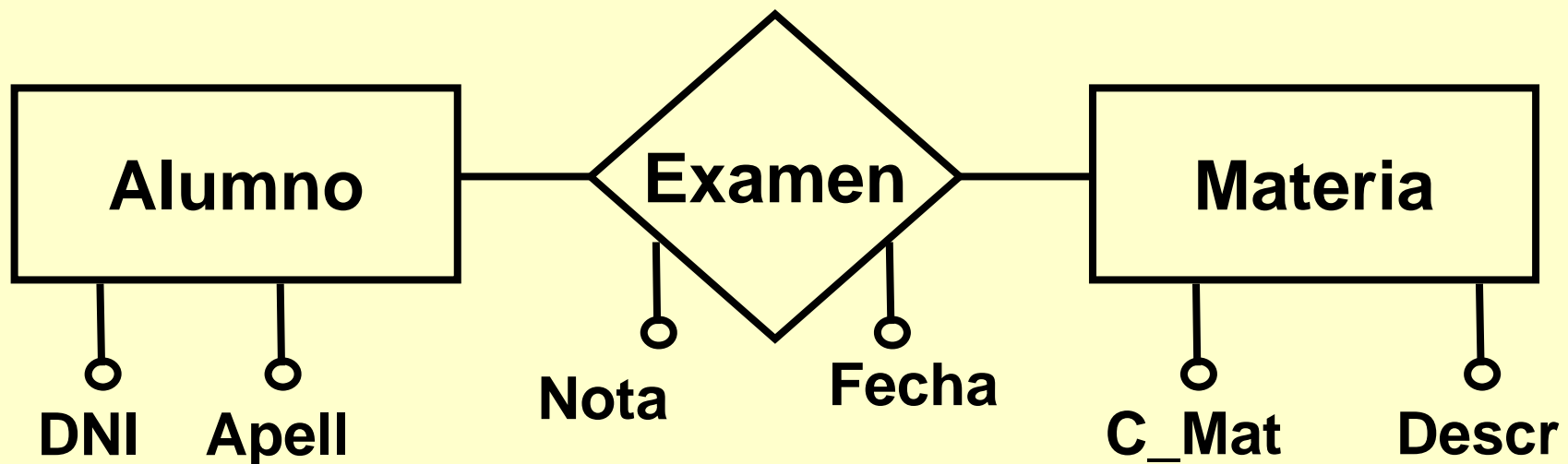


Nota y Fecha no caracterizan a Alumno o a Materia, sino a la relación entre Alumno y Materia

# Identificación de entidades

Tiene que ser posible distinguir un elemento de una entidad de otros.

Una solución viable es encontrar un conjunto de atributos para los cuales la combinación de valores es diferente para cada elemento.





## Traducir entidades y relaciones a tablas

La intuición sugiere traducir cada entidad en una tabla.

### Observaciones:

Es posible encontrar otras soluciones.

Para el mismo atributo, utilizar el mismo identificador.

Cada elemento de la entidad será una fila de la tabla.

## Traducir entidades y relaciones a tablas

La intuición sugiere traducir cada entidad en una tabla.

Alumno ( DNI , Apell )

DNI	Apell
36564289	Pérez
33900561	Muro
36087782	Báez
35678902	Lorenz

Materia ( C\_Mat , Descr )

C_Mat	Descr
952522	Informática I
952873	Álgebra
...	...
...	...

## Traducir entidades y relaciones a tablas

Cada relación es traducida en una tabla, con sus atributos.

Se agrega el identificador de cada entidad vinculada por la relación.

**Restricción de integridad referencial.**

En este ejemplo, el identificador de la tabla está incluido entre los identificadores importados. No siempre es así.

## Traducir entidades y relaciones a tablas

En este ejemplo, el identificador de la tabla está incluido entre los identificadores importados. No siempre es así.

Examen ( DNI , C\_Mat, Nota, Fecha )

DNI	C_Mat	Nota	Fecha
36564289	952522	10	30/07/09
33900561	952522	7	23/07/09
36564289	952873	4	23/07/09
35678902	952522	6	30/07/09

# Grado de las relaciones.

Cantidad de entidades involucradas en la relación.

Relación binaria:

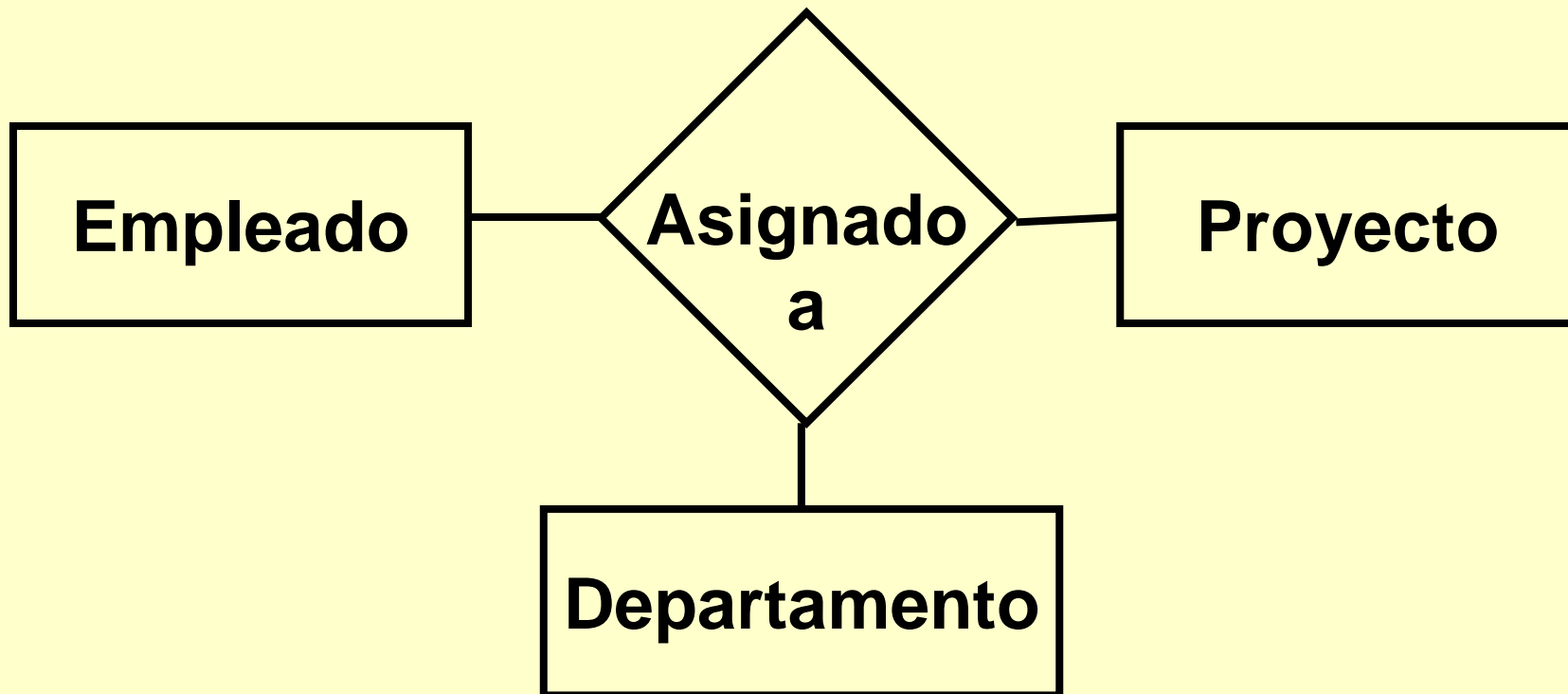
Grado 2



# Grado de las relaciones.

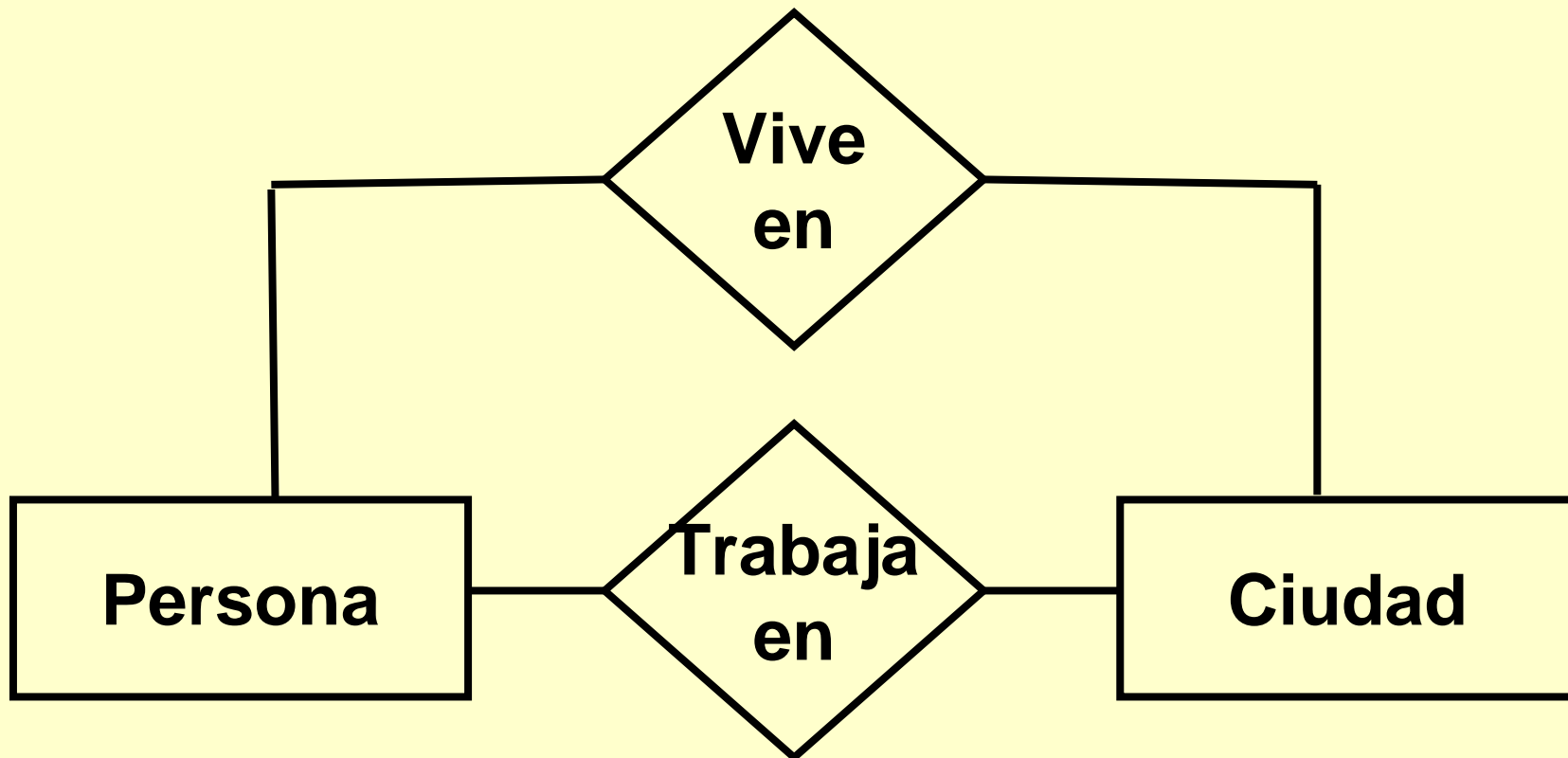
Relación ternaria:

Grado 3



**Dos relaciones vinculan las mismas entidades:**

**Los significados son -obviamente- diferentes, y las relaciones son independientes.**



## **Restricciones en el Modelo E / R**

**Restricciones implícitas: a partir de la semántica de los constructores del modelo.**

**Cada elemento de la relación tiene que relacionarse con elementos existentes de las entidades.**

**Distintos elementos de la relación tienen que relacionarse con distintas combinaciones de elementos de las entidades.**



## **Restricciones en el Modelo E / R**

**Restricciones explícitas: establecidas por el diseñador, a partir de su conocimiento del dominio de aplicación.**

**Cardinalidad.**

**Identificación.**

## **Cardinalidad:**

**Par de valores para cada entidad involucrada en una relación.**

## **Definiciones:**

**\* Cardinalidad máxima: la máxima cantidad de veces que un elemento de una entidad puede participar en elementos de la relación.**

# Restricciones de cardinalidad:

Valores posibles:

Cualquier  
entero

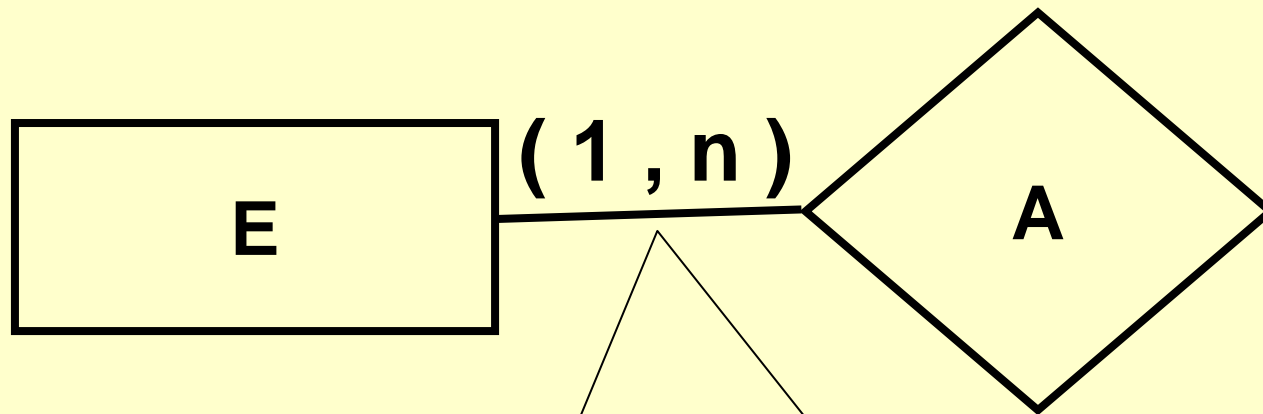
0, 1, etc.

n

Cuando no hay  
restricción

# Restricciones de cardinalidad:

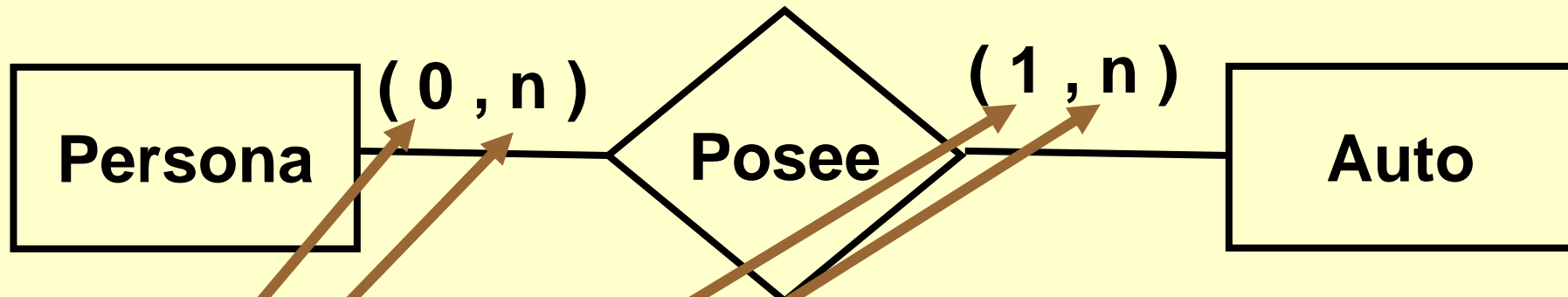
**Ejemplo:**



**Cada elemento de E  
participa al menos una vez  
en A. No hay límite superior.**

# Restricciones de cardinalidad:

Ejemplo:



Significado de la notación (de izquierda a derecha):

- \* Hay personas que no tienen auto.
- \* Una persona puede tener muchos autos.
- \* No hay auto sin dueño.
- \* Muchas personas pueden ser dueñas de un auto.

# Cardinalidad - Casos especiales:

Para relaciones binarias:

Se denomina relación...

- \* **1 a 1**, cuando ambas cardinalidades máximas son 1.
- \* **1 a varios**, cuando un máximo es 1 y el otro es n.
- \* **varios a varios**, cuando ambos son n.

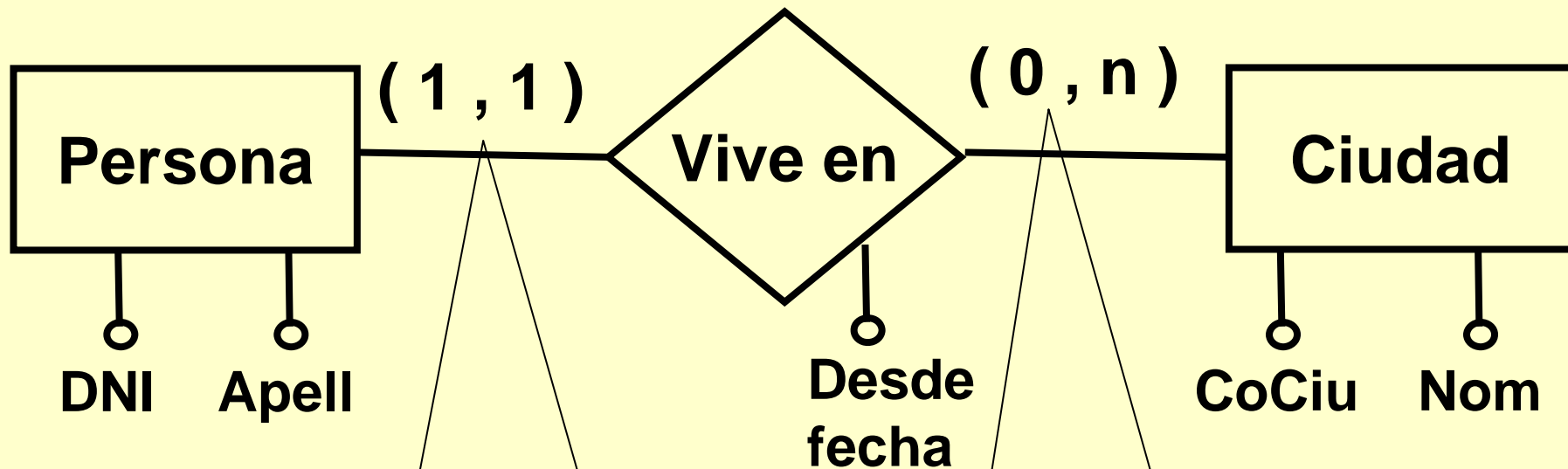
# Cardinalidad

La participación de una entidad en una relación es:

- \* **Opcional**, cuando la cardinalidad mínima es 0.
- \* **Obligatoria**, cuando la cardinalidad mínima es mayor que 0.

# Cardinalidad

## Ejemplo:



En la base de datos,  
se quiere saber en  
qué ciudad vive  
cada persona

En la base interesan  
ciudades en las que  
no vive ninguna de  
las personas



## Restricciones de cardinalidad

- \* **Las restricciones no son “toda la verdad”, pero tienen como objetivo el subconjunto del mundo real de nuestro interés.**

## Traducción relacional

**Ciudad(CoCiu, Nom)**

**Persona(DNI, Apell)**

**Vive\_en (DNI, CoCiu, Desde\_fecha)**

FK(DNI)->Persona

FK(CoCiu)->Ciudad

**¿Clave para  
Vive\_en?**

## **Traducción relacional**

**Según las cardinalidades, la relación es 1 a varios, es decir, cada elemento de Persona puede estar sólo una vez en Vive\_en.**

**En consecuencia, cada fila de Vive\_en tiene que tener un valor distinto en DNI.**

**Por lo tanto, DNI es la clave de Vive\_en.**

## **Traducción relacional**

**En relaciones 1 a varios, la clave de la tabla de relación es aquella de la entidad del “lado del 1”**

# Cardinalidad

## Más ejemplos (I)



**Trabaja\_en ( DNI, CoCiu, Desde\_Fecha )**

**FK(DNI)->Persona**

**FK(CoCiu)->Ciudad**

# Cardinalidad

## Más ejemplos (II)



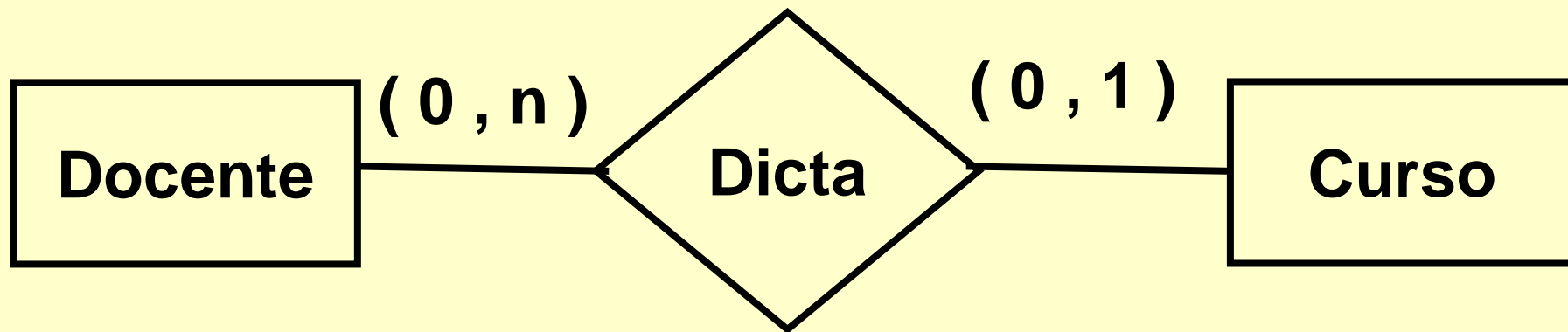
**Asignada ( DNI, IdTes )**

**FK(DNI)->Alumno**

**FK(IdTes)->Tesis**

# Cardinalidad

## Más ejemplos (III)



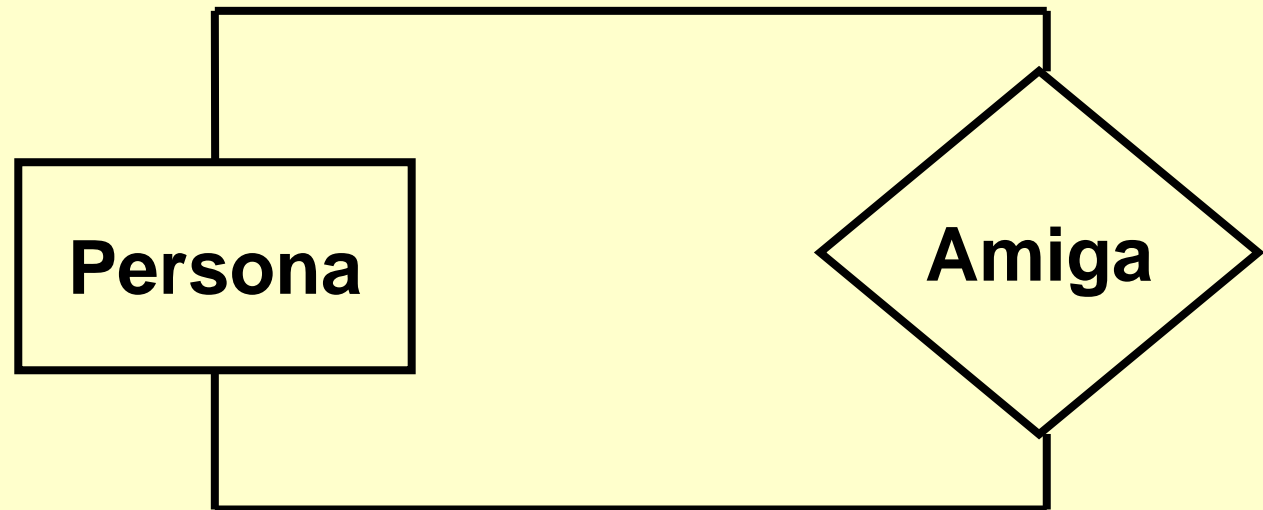
**Dicta ( DNI, NumCur )**

**FK(DNI) -> Docente**

**FK(NumCur) -> Curso**

# Relación cíclica

Vincula a una entidad consigo misma



**Persona ( DNI, Nombre )**

**Amiga ( DNI1, DNI2 )**

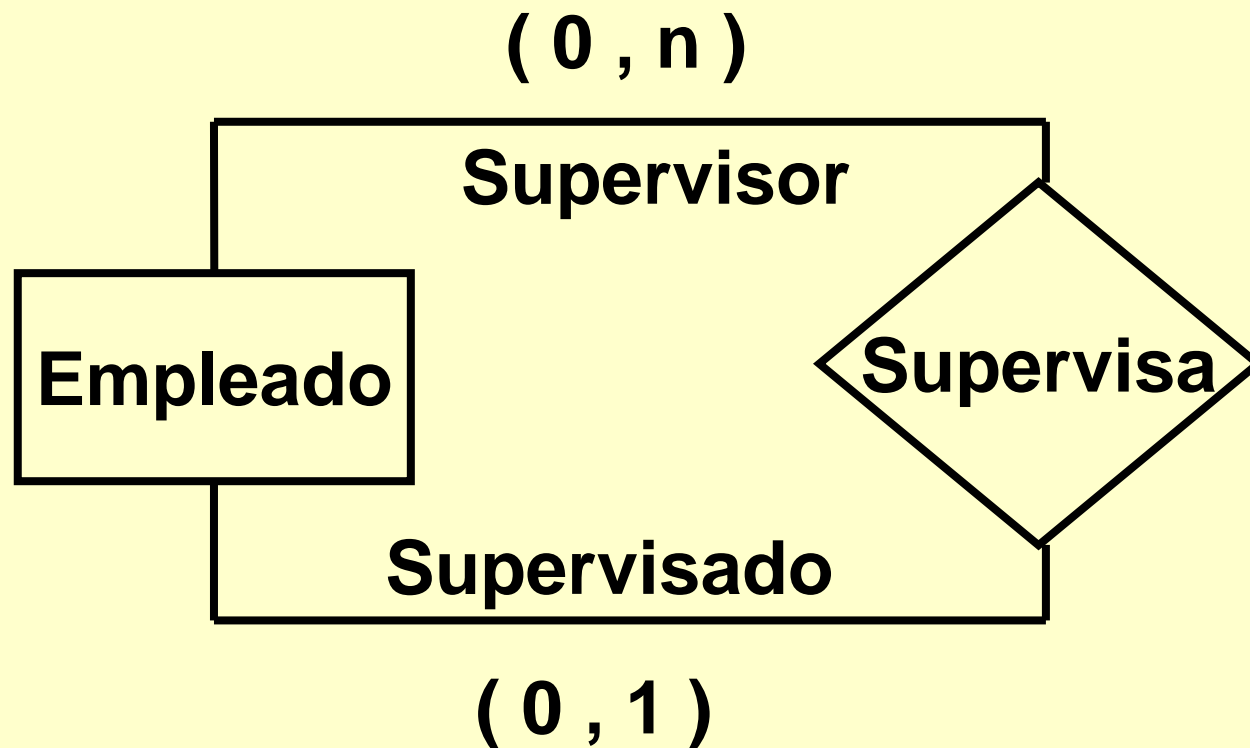
**FK(DNI1) -> Persona**

**FK(DNI2) -> Persona**



# Relación cíclica

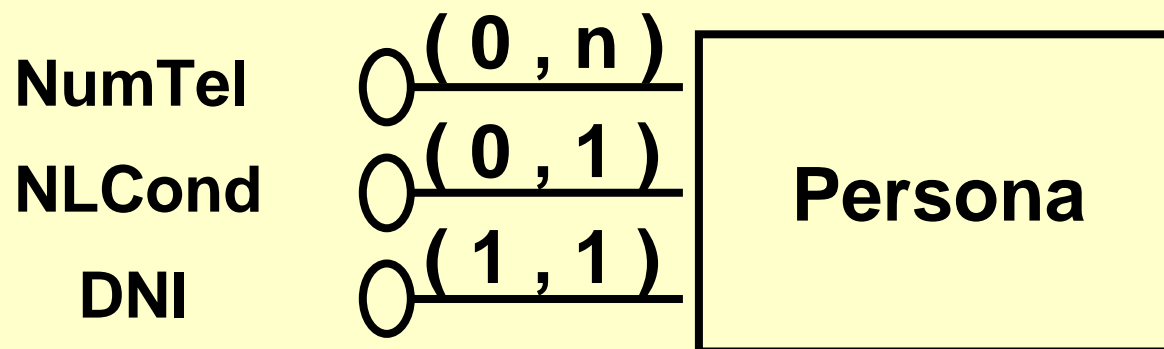
Cuando las cardinalidades no son simétricas, es necesario agregar un “rol” a cada vínculo.



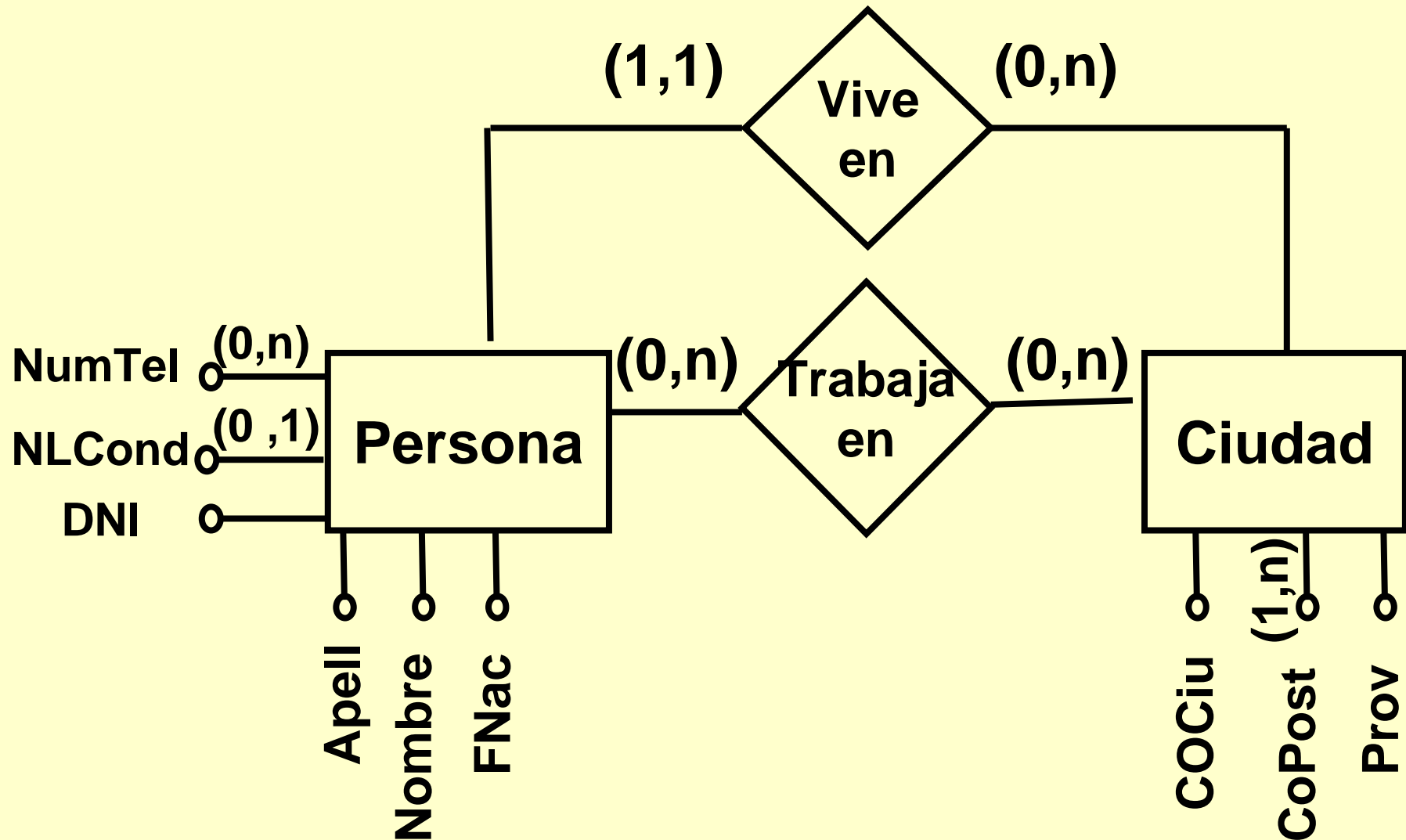
# Restricciones de cardinalidad para atributos

## Valores mínimos y máximos

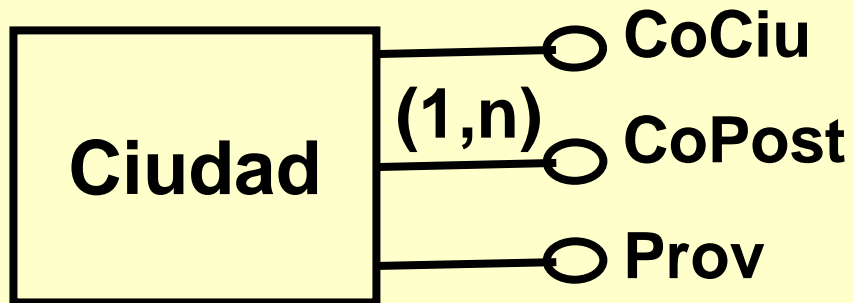
- \* Por defecto se asume (1, 1)
- \* Opcional: cuando la cardinalidad mínima es 0.
- \* Mono-valuado: la cardinalidad máxima es 1.
- \* Multi-valuado: la cardinalidad máxima es mayor que 1.



## Ejemplo con cardinalidades:

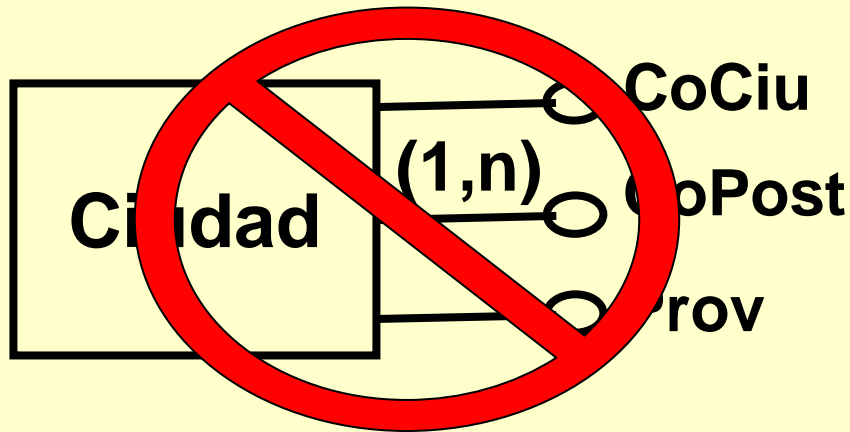


## Traducción de atributos multivaluados:

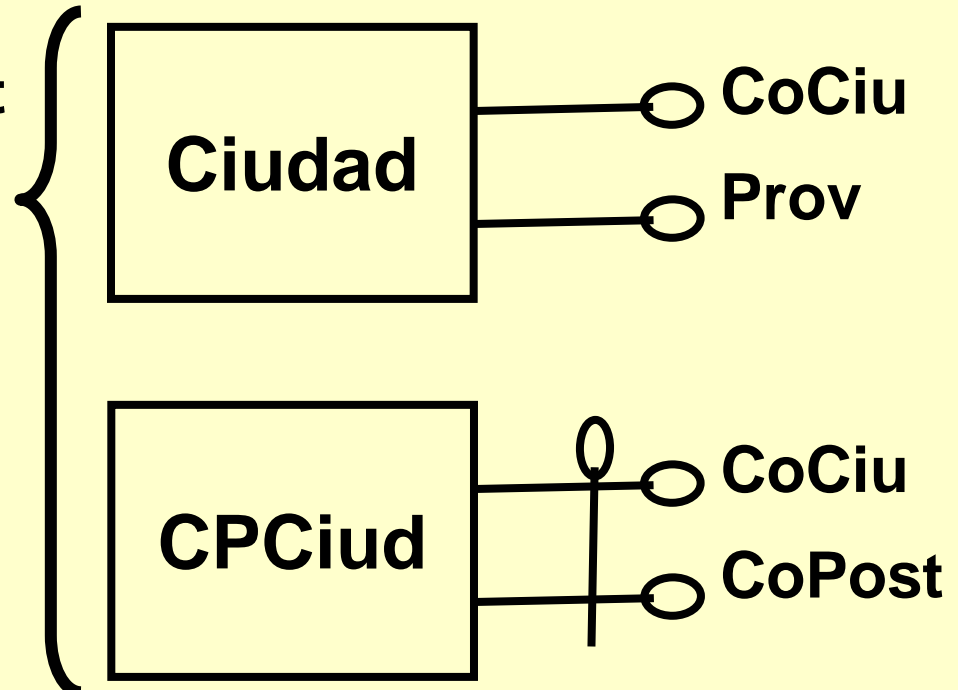


**La existencia de atributos multivaluados implica la existencia de NULLS.**

## Traducción de atributos multivaluados:



**Los atributos multivaluados obligan a modificar el esquema.**



**Ciudad (CoCiu, Prov)**

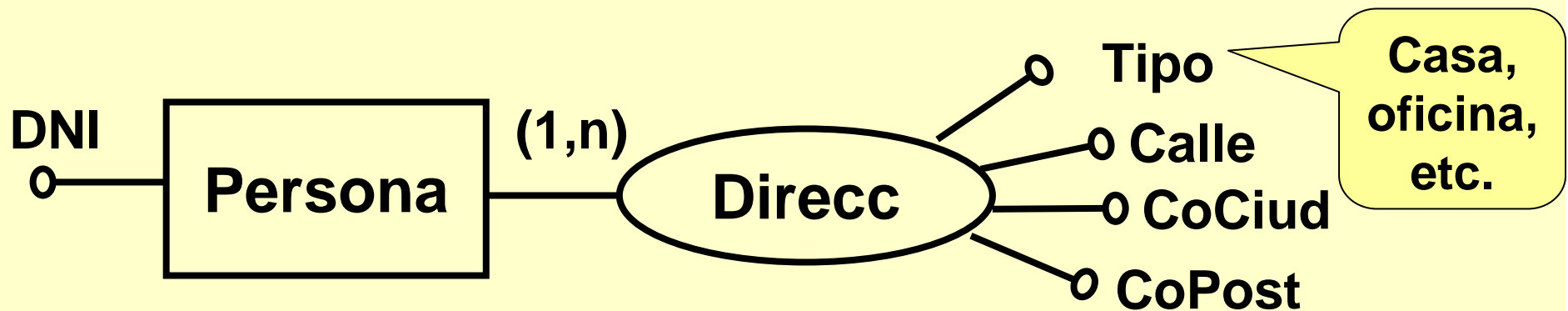
**CPCiud(CoCiu, CoPost)**

**FK(CoCiu) -> Ciudad**

## Atributos compuestos:

### Agregado de atributos simples

Quando existen fuertes afinidades entre atributos simples con respecto a significado y uso.



**Persona (DNI, ...)**

**Direcc(DNI, Ndir, Tipo, Calle, CoCiud, CoPost)**

**FK(DNI) -> Persona**

Distingue entre distintas direcciones

## Identificadores

Un identificador permite distinguir los elementos de una entidad.

**El identificador tiene que ser mínimo, es decir, no más grande que lo necesario.**

## Identificadores

El identificador puede ser:

### **Interno:**

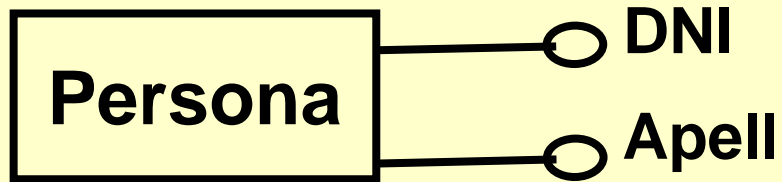
Hay atributos que permiten la identificación.

### **Con componentes externos:**

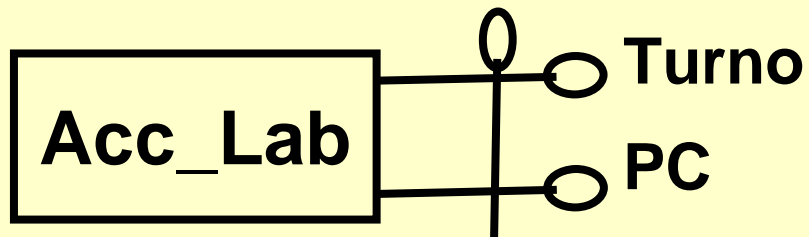
La identificación se completa con una o más entidades vinculadas.



## Identificadores internos y externos

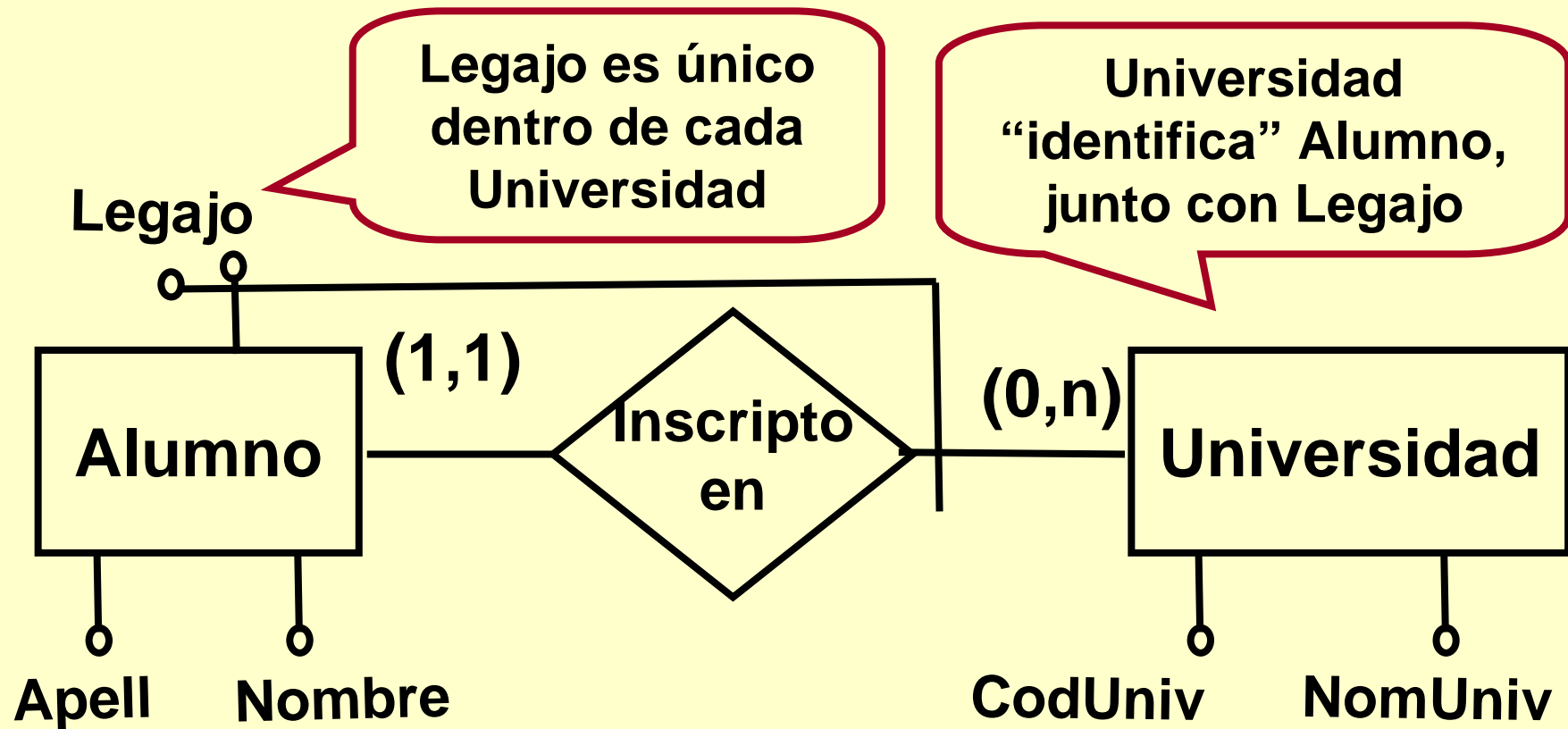


**Identificación simple interna**



**Identificación compuesta interna**

## Identificadores internos y externos



**Identificación compuesta externa**

## Identificación

Cada componente del identificador tiene que ser mono-valuado y obligatorio.

(De otro modo la identificación podría ser incompleta o ambigua).

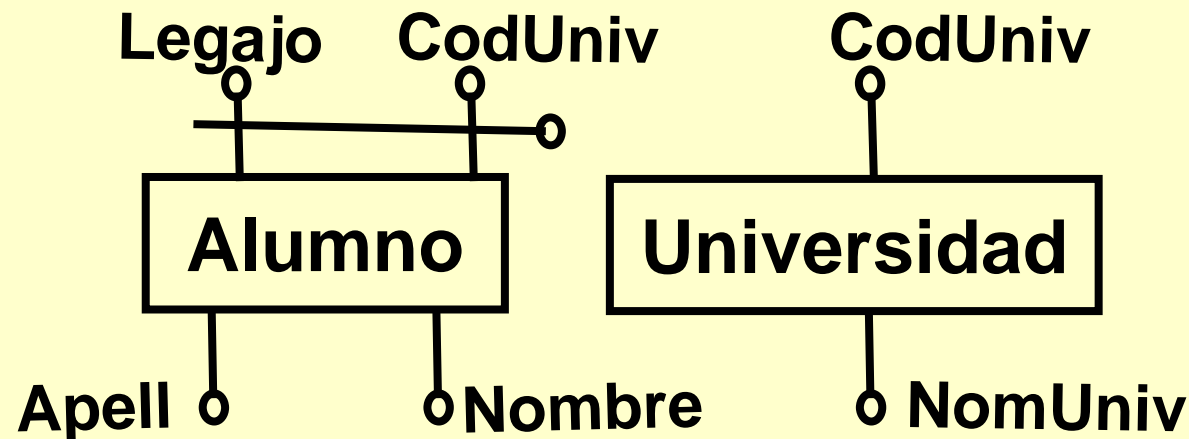
**Toda información sobre identificación es una importante restricción de integridad.**

**A veces los identificadores compuestos son reemplazados por “claves sustitutas”.**

## Traducción de identificación externa

La asociación que aporta la identificación externa es reemplazada por el identificador de la entidad identificadora.

Es posible hacerlo, porque la cardinalidad es (1,1).



Universidad(CodUniv,NomUniv)

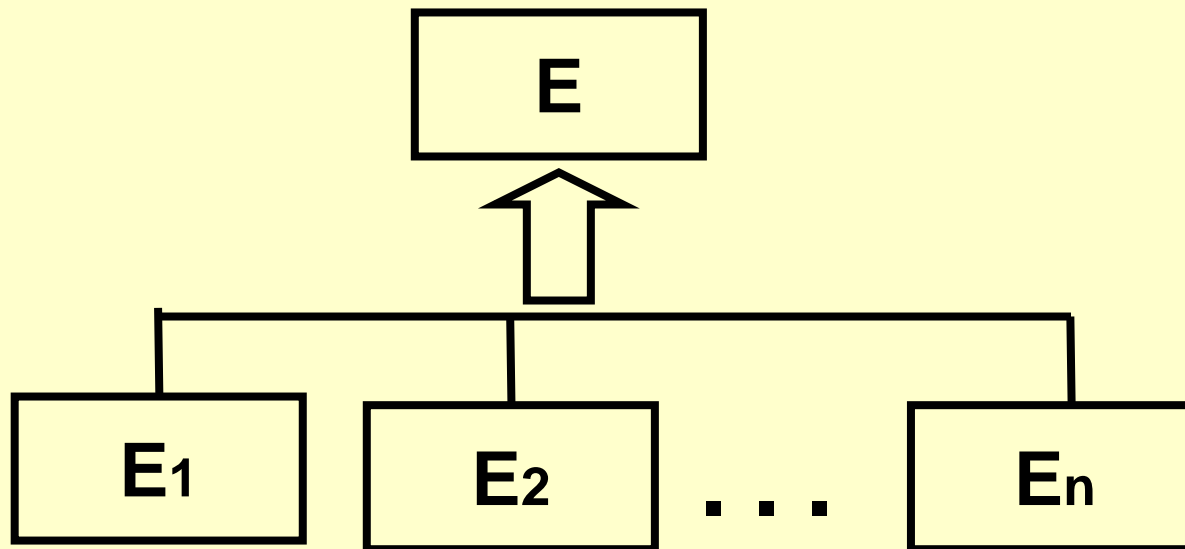
Alumno(CodUniv,Legajo,Apell,Nombre)

FK(CodUniv) -> Universidad

## Jerarquías de generalización

$E$  es una generalización de  $E_1, E_2, \dots, E_n$  si cualquier elemento de cualquiera de las entidades  $E_1, E_2, \dots, E_n$  es también un elemento de  $E$ .

$E_1, E_2, \dots, E_n$  son especializaciones de  $E$ .

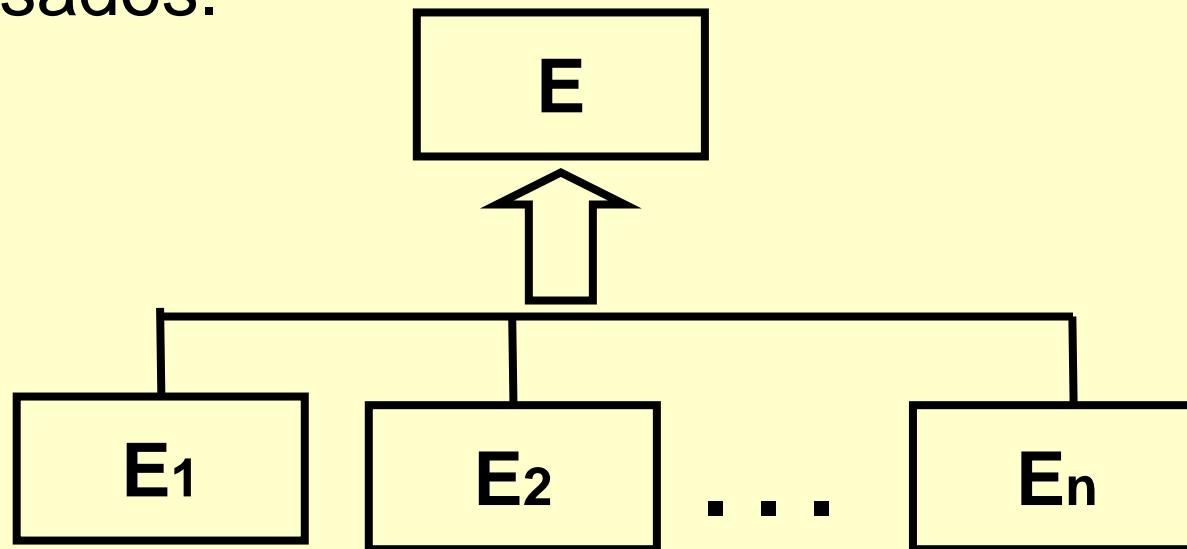


## Jerarquías de generalización

Los atributos de  $E$  son heredados implícitamente por  $E_1, E_2, \dots, E_n$

- \* Cada elemento de  $E_1, E_2, \dots, E_n$  tiene los atributos de  $E$ .

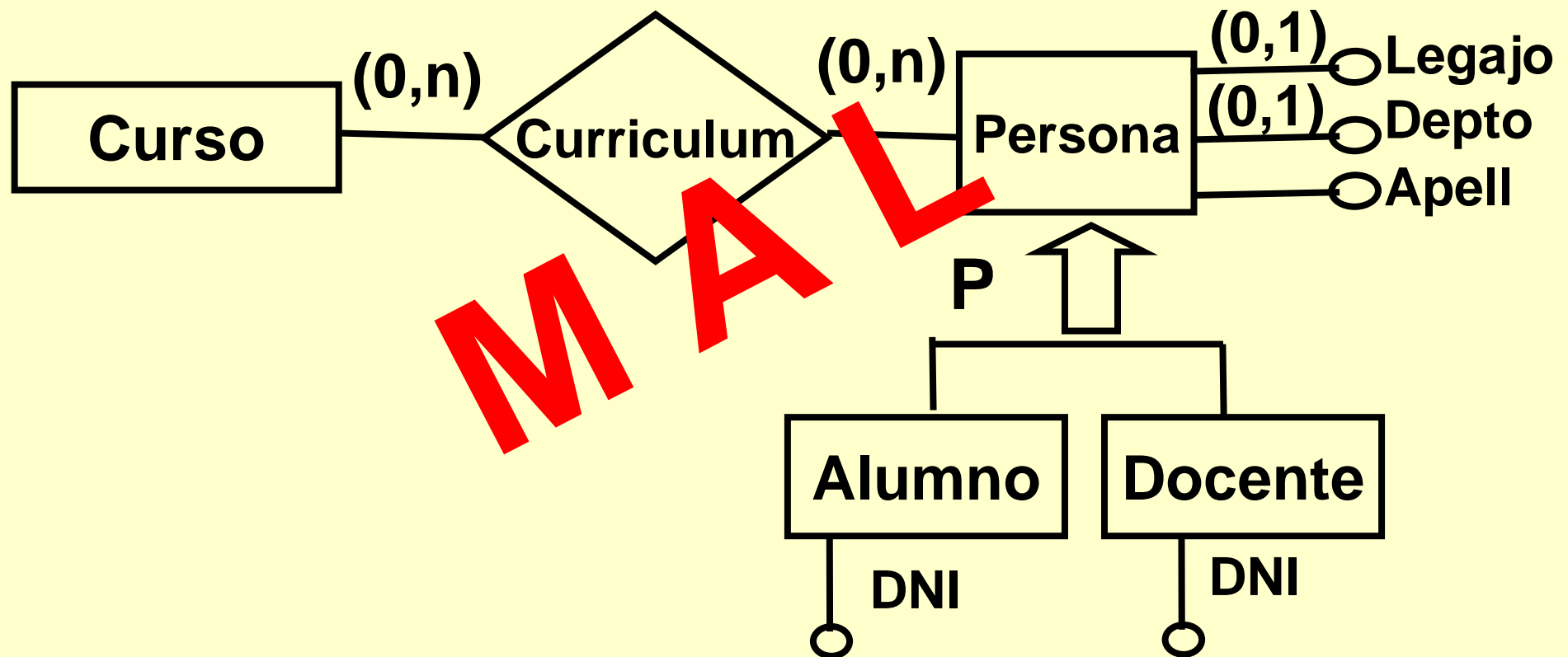
- \* Los esquemas implícitos no requieren ser expresados.



## Herencia de atributos

Los atributos se expresan en la entidad más genérica en la que son obligatorios.

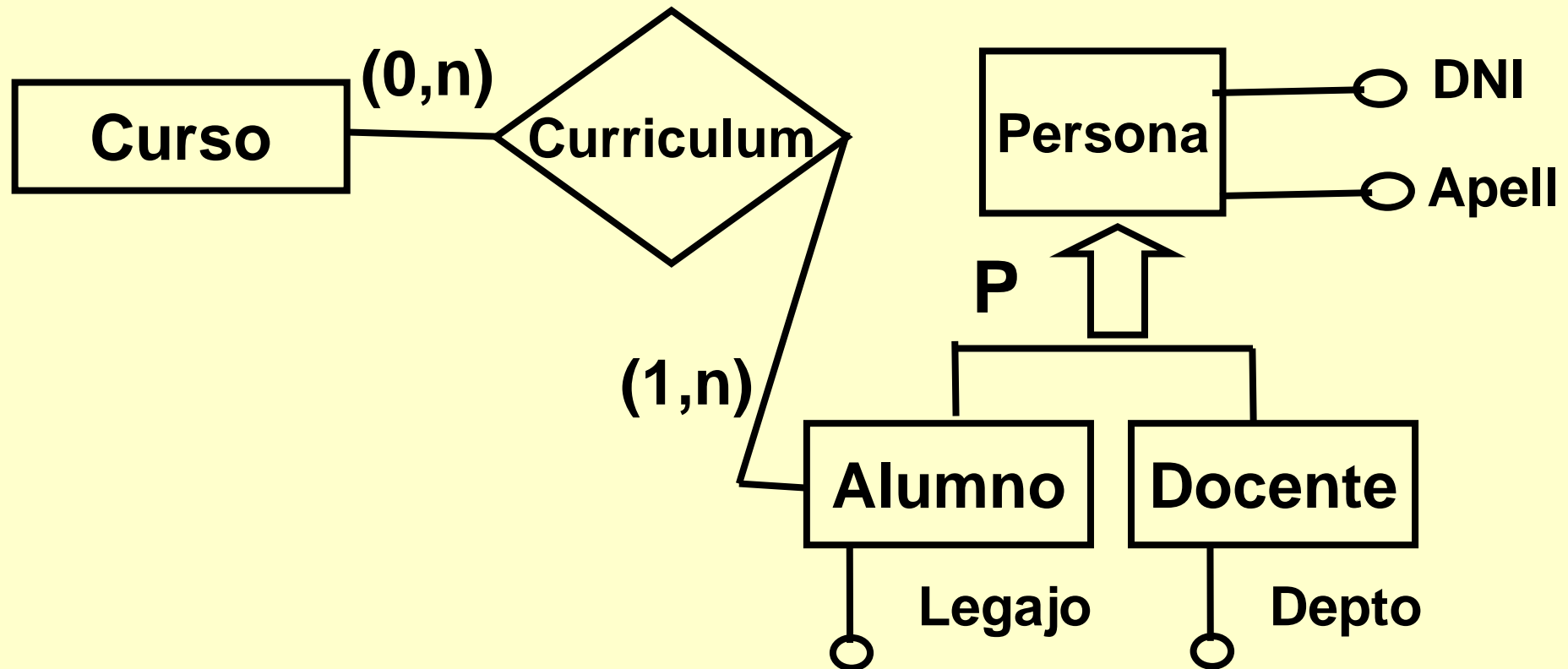
Lo mismo ocurre para relaciones.



## Herencia de atributos

Los atributos se expresan en la entidad más genérica en la que son obligatorios.

Lo mismo ocurre para relaciones.





## Traducción de jerarquías

### Ingenua

Una tabla para cada entidad.

La especialización no tiene identificador.

**Se copia el identificador  
de la generalización en  
las especificaciones.**

**Restricción de clave externa**

## Traducción de jerarquías

### Sintaxis

**Curso(Ncur, Nombre)**

**Persona(DNI, Apell)**

**Alumno(DNI, Leg)**

**Fk(DNI) → (Persona)**

**Docente(DNI, Depto)**

**Fk(DNI) → (Persona)**

**Curriculum(DNI, Ncur)**

**Fk(DNI) → (Alumno)**

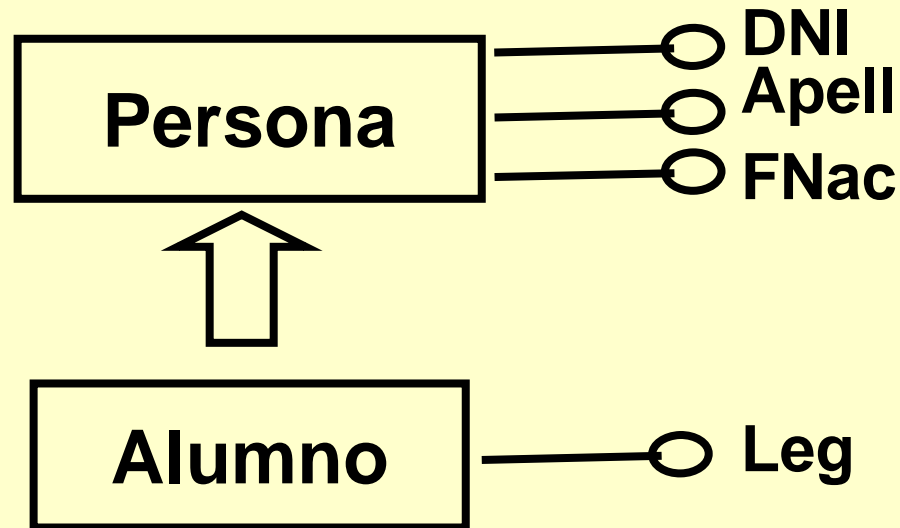
**Fk(NCur) → (Curso)**

**La clave  
externa  
tiene que  
referir a la  
entidad  
más  
cercana**

## Jerarquías de generalización

### Subconjunto

Caso especial con  
especialización única



Alumno hereda  
los atributos de  
Persona

Cada alumno es también  
una persona

## Clave principal

Es bastante habitual la existencia de más de un identificador.

**Por razones prácticas, uno de ellos ha de ser elegido como *clave principal* .**

**La clave primaria permite un acceso más eficiente.**

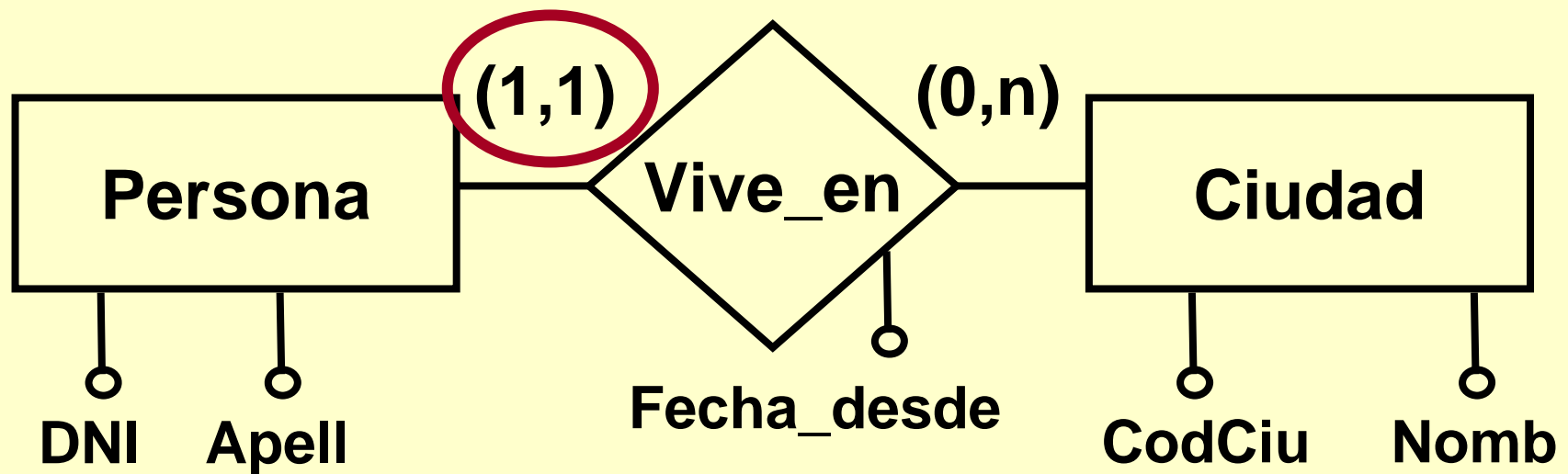
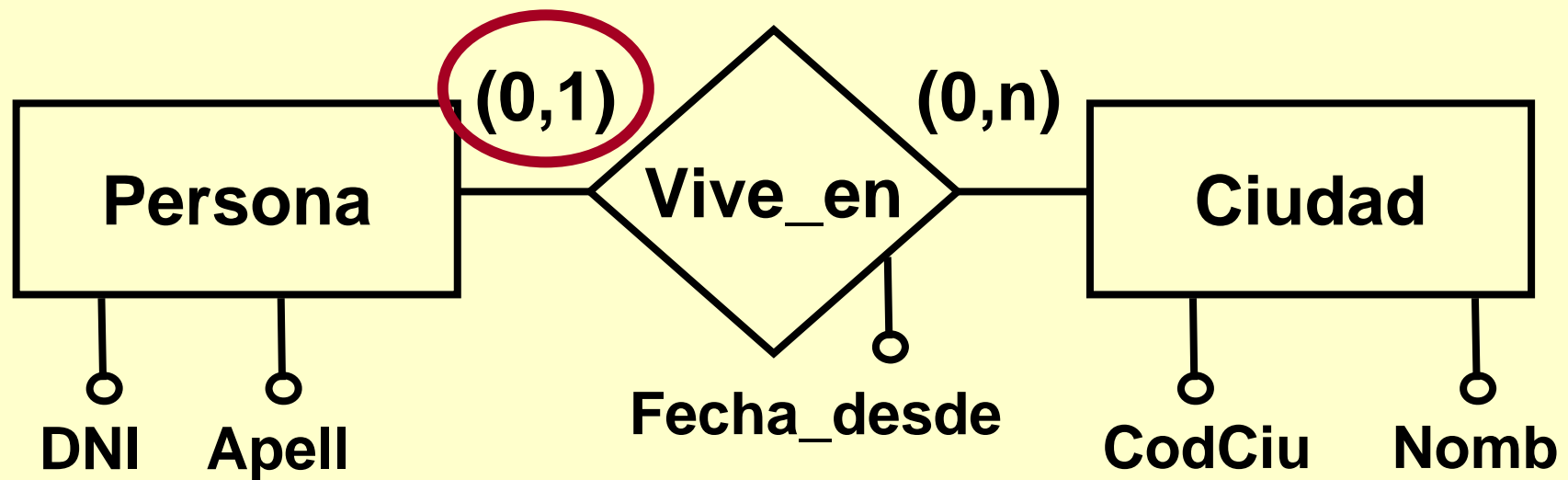
**Se la utilizará para claves externas y uniones relacionales (join).**

## Clave principal

### Reglas prácticas:

- \* **Elegir como clave principal el identificador más utilizado para acceso de un elemento.**
- \* **Preferir los identificadores simples.**
- \* **Preferir identificadores internos.**

# Ejemplo



## Ejemplo

**Los dos esquemas previos se traducen en el mismo esquema relacional.**

**No se preserva la semántica de la relación opcional.**

## Ejemplo

### Sintaxis

**Persona(DNI, Apell)**

**Ciudad(CodCiu, Nomb)**

**Vive\_en(DNI,CodCiu,Fecha\_desde)**

**Fk(DNI) → (Persona)**

**Fk(CodCiu) → (CodCiu)**



## Modelo E / R

# ¿Por qué es útil?

- \* **Es más expresivo que el esquema relacional.**
- \* **Puede ser satisfactoriamente utilizado para:**
  - **Documentación.**

La representación gráfica puede ser rápidamente comprendida por cualquiera.
  - **Ingeniería inversa.**

Un esquema de base de datos puede ser descrito en E/R para análisis y, posiblemente, para reingeniería.
  - **Integración.**

Proporciona una visión sintética capaz de representar sistemas heterogéneos.

# Limitaciones

- \* **El modelo E/R no es capaz de capturar todo.**
  - Los nombres a veces no son suficientes para lograr una comprensión completa.
  - No todas las restricciones de integridad pueden ser expresadas con E/R

*“... para poder rendir la materia X, es necesario cumplir las correlatividades”*
- \* **La fase de diseño tiene que integrar el diagrama E/R con una adecuada documentación escrita.**

## Resumen

- \* El modelo E/R puede ser utilizado para diseñar bases de datos.**
- \* Los conceptos principales son: entidad, relación, atributo. El esquema se enriquece entonces con identificadores, cardinalidades y jerarquías.**
- \* Es necesaria documentación de apoyo para adicionar más semántica.**

## Opciones alternativas

- \* **Es posible realizar un diseño lógico standard automáticamente.**
- \* **En muchos casos, existe la posibilidad de soluciones alternativas.**
  - **Tienen todas el mismo “significado”, es decir pueden ser pobladas por la misma información.**
  - **Difieren con respecto a la performance, dependiendo del tipo de operación a realizar**

## Opciones alternativas

Es necesario tener en cuenta:

- \* **Tipo de información y frecuencia de operaciones.**
- \* **Información acerca del número de elementos en cada entidad.**
- \* **Información acerca de proporciones.**
  - **Proporción de null values.**
  - **Proporción de elementos de entidad no participantes en relación**

## Opciones alternativas

### **Algunas reglas prácticas:**

- \* Las propiedades lógicas han de ser preservadas, independientemente de la eficiencia.**
- \* Aquellos atributos a ser utilizados juntos frecuentemente, serán guardados en la misma tabla.**
- \* Reducir el porcentaje de null values.**

## **Problemas habituales en diseño conceptual**

### **Algunas reglas prácticas:**

- \* Hay ciertos “patrones de diseño” encontrados con frecuencia en casos prácticos.**
- \* No hay algunas soluciones generales, pero algunas “buenas prácticas” pueden orientar al diseñador.**

## Ejemplo

### Áreas de la ciudad

**La ciudad está dividida en tres áreas (*microcentro, macrocentro y suburbios*). Para cada área hay un índice distinto a aplicar a la tasación de propiedades.**

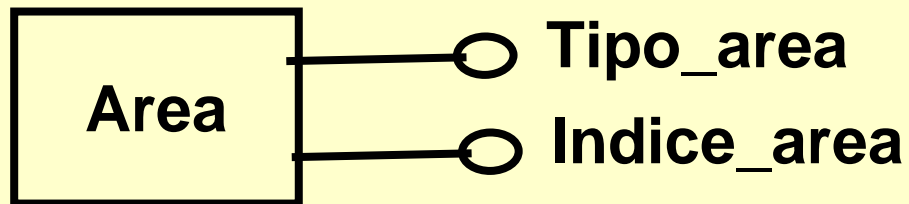


## Ejemplo

# Áreas de la ciudad

**Se trata de un caso de enumeración.**

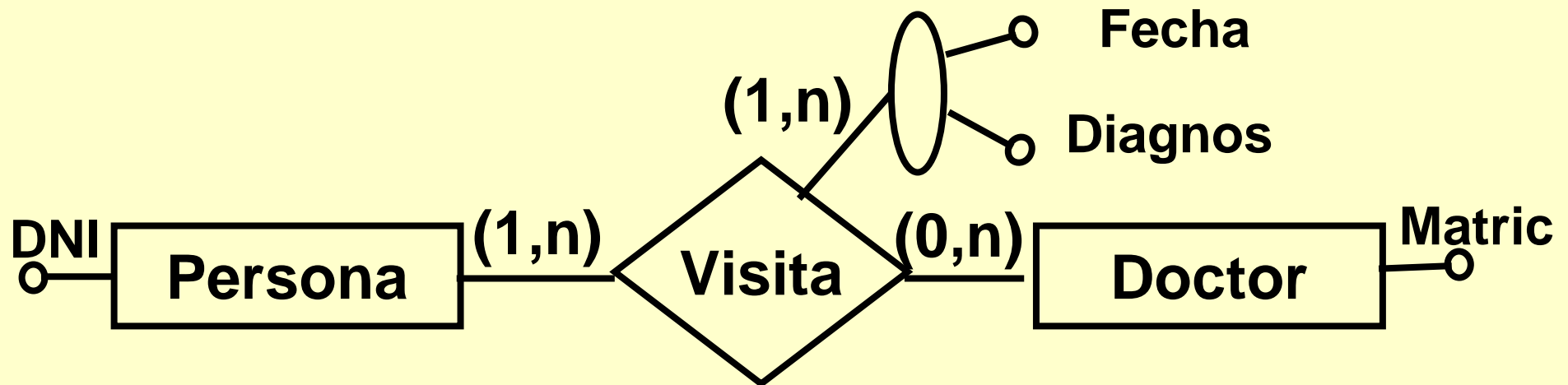
**La enumeración se transforma en un atributo identificador, sin jerarquía.**



## Ejemplo

# Repeticiones en el tiempo

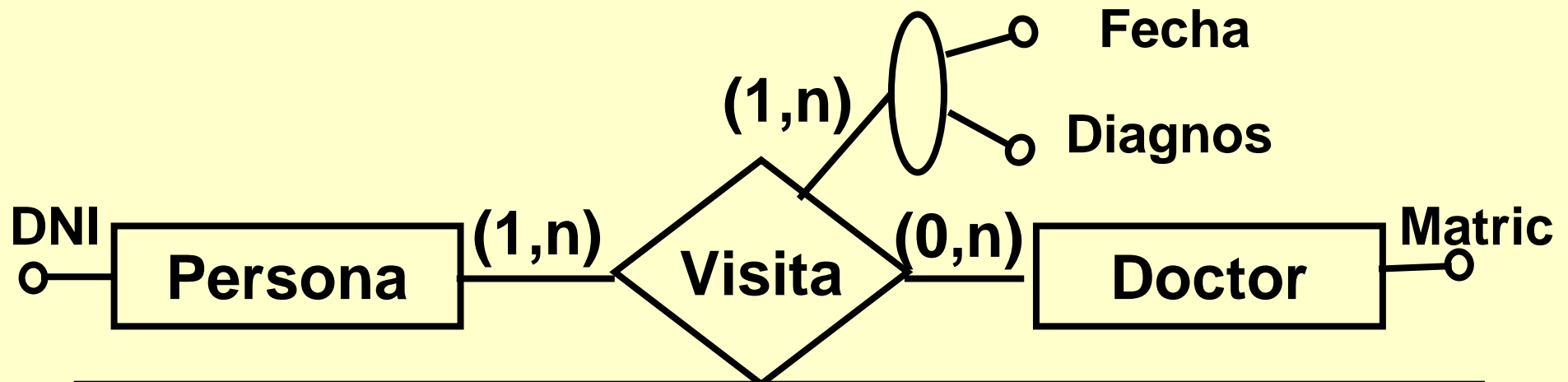
Una persona puede ser atendida por diferentes médicos, pero no más de una vez por día por el mismo médico.



## Ejemplo

# Repeticiones en el tiempo

La fecha tiene que ser parte del identificador de la relación.



Pero las relaciones son identificadas por entidades...

**Fin de la presentación**

Referencias:

\* y otros